A Process-theoretic Approach to Supervisory Control of Interactive Markov Chains

J. Markovski

Department of Mechanical Engineering, Eindhoven University of Technology
Den Dolech 2, 5612MH, Eindhoven, Netherlands

j.markovski@tue.nl

Abstract

We propose a process-theoretic approach to supervisory control of stochastic discrete-event systems with unrestricted nondeterminism. The model of choice is termed Interactive Markov Chains, a natural semantic model for stochastic variants of process calculi and Petri nets. We employ a stochastic extension of the behavioral preorder partial bisimulation to capture the notion of controllability and preserve correct stochastic behavior. The stochastic behavior is preserved up to lumping of Markovian delays. To synthesize a supervisor, we abstract from the stochastic behavior and show that the obtained supervisor is suitable for the original system as well.

Keywords

Supervisory Control Theory; Interactive Markov Chains; Partial Bisimulation Preorder; Controllability; Supervisor Synthesis

Introduction

Development costs for control software of high-tech systems are constantly increasing due to ever-rising complexity of the machines and demands for better quality, safety, and performance. Traditionally, the control requirements are formulated in informal documents and translated into control software, followed by code validation and testing. However, this iterative process becomes time-consuming due to frequent changes and ambiguity of the specification documents. This issue gave rise to supervisory control theory developed by Ramadge and Wonham (1987), where supervisory controllers that coordinate discrete-event system behaviour are synthesized automatically based on formal models of the hardware and the control requirements.

The supervisory controller observes machine behavior by receiving signals from ongoing activities and sends feedback in terms of control signals about allowed activities. We work under the standard assumption that the supervisory controller reacts sufficiently fast on machine input. In this case this feedback loop can be modeled as a pair of synchronizing processes, cf. Cassandras and Lafortune (2004). We refer to the model of the machine as plant, which is restricted by synchronization with the model of the controller, known as a supervisor.

Model-based Systems Engineering

We structure the modelling process in a model-based systems engineering framework depicted in Fig. 1, which extends previous proposals of Schiffelers et al. (2009), Markovski et al. (2010), and Markovski (2011b). Following the model-based methodology, domain engineers initially specify the functionality of the desired controlled system. This leads to a design, developed by the domain and software engineers together. This design defines the modeling level of abstraction and control architecture and it results in informal specifications of the plant, the control, and the performance requirements. Next, the plant and control requirements are modeled in parallel. We synthesize a supervisor based on the abstracted version of the plant, which is coupled with the original variant of the plant to obtain the complete stochastic supervised behavior of the system. Note that the control requirements specify only desired safety functional properties of the system.

The succeeding steps validate that the control is meaningful, i.e., desired functionalities of the controlled plant are preserved. This step involves stochastic verification of the supervised plant based on the model of the performance requirements, e.g. in the vein of Baier et al (2010), or validation by simulation, as proposed in Schiffelers et al. (2009). If validation fails, then the control requirements are remodeled, and sometimes a complete revision proves necessary. Finally, the control software is generated automatically based on the validated models, shifting the focus of software engineers from coding to modeling.

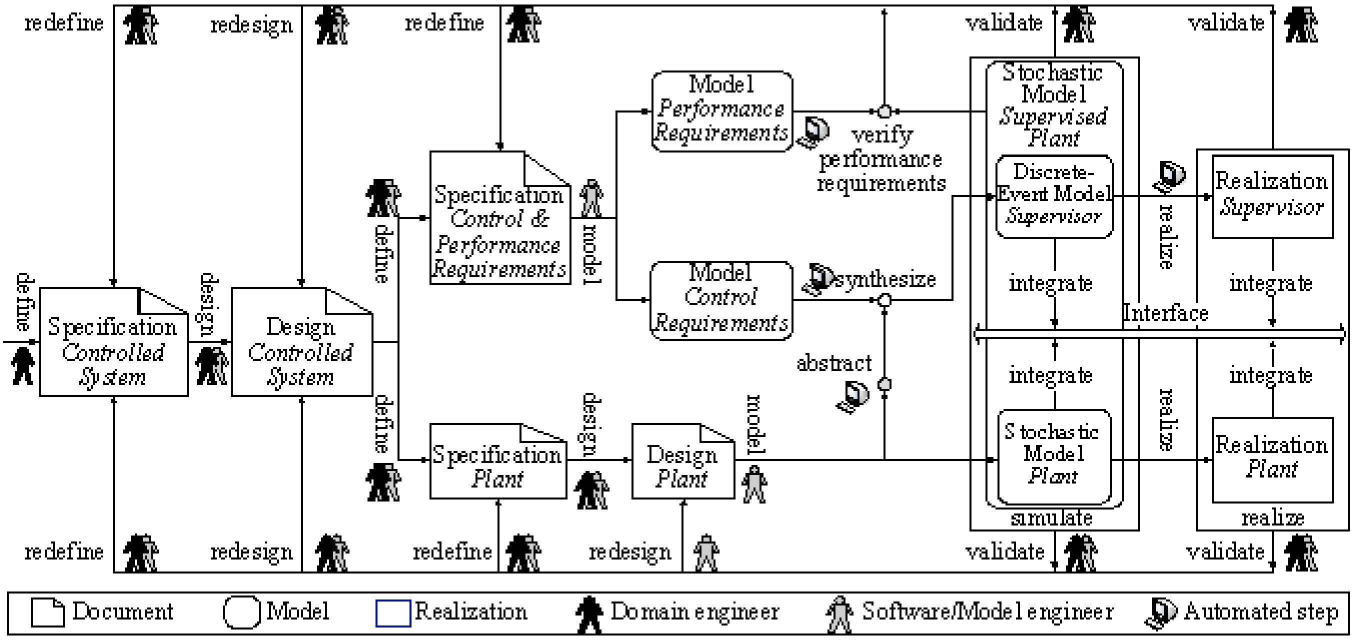

FIG. 1 PROPOSAL FOR A SYNTHESIS-CENTRIC MODEL-BASED SYSTEMS ENGINEERING FRAMEWORK

Motivation and Contributions

Our main motivation for the proposed framework are recent advances in verification of stochastic properties of dynamic systems, summarized in Baier et al. (2010) and Hermanns and Katoen (2010). These techniques employ probabilistic or stochastic extensions of temporal logics to conveniently specify performance and dependability guarantees for Markov (reward) processes, see Howard (1971), in a modular and flexible manner. They provide a unified framework for checking satisfiability of both qualitative and quantitative specifications.

To support supervisory control of (nondeterministic) stochastic discrete-event systems, we employ the process-theoretic model of Interactive Markov Chains (IMCs), proposed in Hermanns (2002). IMCs uniquely couple labeled transition systems, a standard model which captures nondeterministic discrete-event behavior, cf. Baeten et al. (2010), with continuous-time Markov chains, cf. Howard (1971), the most prominent performance and reliability model. The extension is orthogonal, arbitrarily interleaving exponential delays with labeled transitions. In Hermanns and Katoen (2010), it is argued a natural semantic model for stochastic process calculi, see Hermanns et al. (2002) and (generalized) stochastic Petri nets, see Ajmone Marsan et al. (1995).

Our contribution is a process-theoretic approach to supervisory control of IMCs that captures the central notion of controllability by means of a behavioral relation. Controllability defines the conditions under which a supervisor exists such that the control requirements are achievable by synchronizing it with the plant. By employing behavioral relations to define controllability, we can provide for separation of concerns, i.e., we can separate the synthesis procedure from the analysis of the stochastic behavior. To this end, we abstract from the stochastic behavior of the system, generate a supervisor for the abstracted plant, which is compatible with the original stochastic plant, and employ this supervisor to obtain the stochastic supervised plant. The abstraction is supported by the memoryless property of the Markovian delays, cf. Howard (1971), which enables us to treat them syntactically as a special type of dicrete-events in the parallel composition.

Related Work

Supervisory control theory is traditionally language-based, cf. Ramadge and Wonham (1987) and Cassandras and Lafortune (2004). Early process-theoretic approaches employ failure semantics, e.g., Heymann and Lin (1998) and Overkamp (1997). The use of refinement relations to relate the supervised plant, given as a desired control specification to be achieved, to the original plant was studied in Overkamp (1997), Rutten (2000), and Zhou et al. (2006). The approach of Baeten et al. (2011) proposed to employ the behavioral preorder partial bisimulation as a suitable behavioral relation to define controllability.

Regarding optimality, Markov decision processes are an extension with control features that enables a choice between several possible future behaviors, cf.

Howard (1971). The control problem is scheduling of the control actions by dynamic programming, e.g., see Bertsekas (2007). Stochastic games problem variants employing probabilistic temporal logics are also emerging, like Baier et al. (2005). The supervisory control community typically extends the original approach with quantitative features, like costs, probabilities, and stochastic delays. These extensions are language-based, e.g., see Lawford and Wonham (1993), Garg et al. (1999), Kwong and Zhu (1995), or Kumar and Garg (1998).

We aim to exploit the strengths of the approaches from the different communities by employing traditional techniques to first synthesize a supervisor that will conform to the qualitative control requirements. Then, we look for optimal supervision that respects the performance requirements, enforced by stochastic model checking techniques. What will enable us to apply both techniques is the choice of the underlying process-theoretic model of IMCs.

The proofs of the theorems in this paper are given in Markovski (2011b) and the references therein.

Interactive Markov Chains

IMCs are extensions of labeled transition systems with Markovian delays labeled on the transitions by the rates of the exponential distributions.

Def. 1 An IMC is a tuple $I = (S, s_0, A, l, m, t)$, where S is a set of states with initial state $s_0 \in S$, A is a set of action labels, $l \subseteq S \times A \times S$ is a set of labeled transitions, $m \subseteq S \times R \times S$ is a set of Markovian transitions, and $t \subseteq S$ is a successful termination predicate.

An IMC becomes a labeled transition system if there does not exist $s \in S$ such that $(s, p, s') \in m$. It becomes a conventional Markov chain if there does not exist $s \in S$ such that $(s, a, s') \in l$ and $t = \emptyset$. Labeled transitions are interpreted as delayable actions in process theories, see Baeten et al. (2010), i.e., an arbitrary amount of time can pass in a state before an outgoing labelled transition is taken.

Semantics

Intuitive interpretation of a Markovian transition $(s, p, s') \in m$ is that there is a switch from state s to state s′ within a time delay with duration $d > 0$ with probability $1 - e^{-pd}$, i.e., the Markovian delays are distributed according to a negative exponential distribution parameterized by the label p. By rate (s, s′), we denote the rate to transit from s to s′, i.e., rate (s, s′) $= \sum\{ p \mid (s, p, s') \in m \}$. By rate(s, C) we denote the exit rate of s to some subset $C \subseteq S$ given by rate(s, C) = $\sum\{ \text{rate}(s,s') \mid s' \in C \}$.

If a given state s has multiple outgoing Markovian transitions, then there is probabilistic choice between these transitions and the probability of transiting to s′ following a delay with duration $d > 0$ is given by $\frac{\text{rate}(s, s')}{\text{rate}(s, S)}(1 - e^{-\text{rate}(s, S)\, d})$. Roughly speaking, a discrete probabilistic choice is made on the winning transition with shortest exhibited duration to a candidate outgoing state with a duration determined by the total exit rate of the origin state. In case of mixed labeled and Markovian outgoing transitions, we have a nondeterministic choice followed by a passage of time in which no Markovian has expired. The successful termination option predicate denotes states in which we consider the modeled process to be able to successfully terminate.

Synchronization

The synchronization of IMCs is defined by synchronizing labelled transitions with the same labels, whereas the other transitions and Markovian delays are interleaved. It also merges labeled transitions in a lock-step manner, i.e., two synchronizing transitions must be merged if they have the same labels. Since labeled transitions can delay arbitrarily, they can be consistently interleaved with the Markovian delay, one of the greatest advantages of this model, see Hermanns and Katoen (2010).

Def. 2 Given two IMCs $I_1 = (S_1, s_1, A_1, l_1, m_1, t_1)$ and $I_2 = (S_2, s_2, A_2, l_2, m_2, t_2)$, their parallel composition is defined by $I = I_1 \mid I_2 = (S, s_0, A, l, m, t)$, where $S = S_1 \times S_2$, $s_0 = s_1 \times s_2$, $A = A_1 \cup A_2$, $s = (s', s'') \in t$, if $s' \in t_1$ and $s'' \in t_2$, $((s', s''), p, (s''', p, s'')) \in m$ if $(s', s''') \in m_1$, $((s', s''), p, (s', s''')) \in m$ if $(s'', p, s''') \in m_2$, and $((s', s''), a, (s''', s'')) \in l$ if $(s', a, s''') \in l_1$ and $a \in A_1 \setminus A_2$, $((s', s''), a, (s', s''')) \in l$ if $(s'', a, s''') \in l_2$ and $a \in A_2 \setminus A_1$, and $((s', s''), a, (s''', s'''')) \in l$ if $(s', a, s''') \in l_1$, $(s'', a, s'''') \in l_2$, and $a \in A_1 \cap A_2$.

Partial Bisimulation

We define the notion of controllability by means of the behavioral preorder termed Markovian partial bisimulation. It is an extension of the notion proposed in Baeten et al. (2011) for stochastic discrete-event systems. It states that some events can only be simulated, whereas a subset of the events needs to be bisimulated, in the sense of Glabbeek (2001). The

Markovian transitions are treated as in Markovian lumping, which is the standard minimization procedure for Markovian processes, cf. Howard (1971) and Hermanns (2002).

We introduce some preliminary notions. Given a relation R, we write R^{-1} for the inverse relation. We note that if R is reflexive and transitive, then so is R^{-1}, whereas $R^2 = R \cap R^{-1}$ is an equivalence. We employ this equivalence to ensure that the exiting Markovian rates to equivalence classes coincide as in the definition for Markovian lumping, cf. Hermanns (2002).

Def. 3 Given two IMCs $I_1 = (S_1, s_1, A_1, l_1, m_1, t_1)$ and $I_2 = (S_2, s_2, A_2, l_2, m_2, t_2)$, a reflexive and transitive relation $R \subseteq S_1 \times S_2$ is a Markovian partial bisimulation with respect to the bisimulation action set $B \subseteq A_1$ if for all $(s, t) \in R$ it holds that

1. $s \in t_1$ if and only if $s \in t_2$;
2. for all $s' \in S_1$ and $a \in A_1$ such that $(s, a, s') \in l_1$, there exists $t' \in S_2$ such that $(t, a, t') \in l_2$ and $(s', t') \in R$;
3. for all $t' \in S_2$ and $b \in B$ such that $(t, b, t') \in l_2$, there exists $s' \in S_1$ such that $(s, a, s') \in l_1$ and $(s', t') \in R$;
4. rate (s, C) = rate (t, C) for all $C \in S_1 \times S_2 / R^2$.

If there exists a partial bisimulation R such that $(s_1, s_2) \in R$, then we say that I_2 partially bisimulated I_1 and we write $I_1 <_B I_2$. If $I_2 <_B I_1$ holds as well, then we write $I_1 =_B I_2$. We omit B, when clear from the context.

If the processes do not comprise Markovian prefixes, then the Markovian partial bisimulation coincides with the preorder of Baeten et al. (2011), which additionally is required to be reflexive and transitive. In that case, $<_\varnothing$ coincides with strong similarity preorder and $=_\varnothing$ coincides with strong similarity equivalence, whereas $=_A$ reduces to strong bisimilarity. If the processes comprise only Markovian prefixes, then the relation corresponds to ordinary Markovian lumping. If the processes comprise both action and Markovian prefixes, and if B = A, then $=_A$ corresponds to strong Markovian bisimulation of Hermanns (2002).

Thm. 1 Markovian partial bisimulation $<_B$ is a precongruence for the synchronization operator for every $B \subseteq A$.

A direct consequence of Thm. 1 is that $=_B$ is a congruence, which enables a full process-algebraic treatment of supervisory control theory for IMCs in the vein of Baeten et al. (2010).

Controllability

We employ Markovian partial bisimulation to define controllability from a process-theoretic perspective. As standard practice, we split the set of actions A into a set of uncontrollable actions U and a set of controllable actions C, such that $U \cup C = A$ and $U \cap C = \varnothing$. The former typically represent activities of the system at hand over which we do not have any control, like sensor observation or user and environment interaction, whereas the latter can be disabled or enabled in order to achieve the behavior given by the control requirements, e.g., interaction with the actuators.

Supervised Plant

The plant is typically given by a set of synchronizing IMCs, which ultimately amount to the IMC $P = (S_P, s_P, A_P, l_P, m_P, t_P)$ that does not have any structural restrictions. We note that for the purpose of this paper, we do not need the modular structure of the plant, which can be employed for more efficient synthesis in some cases, cf. Cassandras and Lafortune (2004). The supervisor, however, achieves the control by observing the discrete events of the plant and it must send unambiguous control signals as feedback. Thus, the supervisor does not comprise any stochastic behavior and it must be a deterministic process. An IMC is deterministic if for every $(s, a, s') \in l$ and $(s, a, s'') \in l$ it holds that $s' = s''$. Now, we have that the supervisor is given by the deterministic IMC $S = (S_S, s_S, A_S, l_S, \varnothing, t_S)$, where the Markovian transition relation m_S is empty. The parallel composition P | S specifies the supervised plant, which models the behavior of the supervised system.

Intuitively, the uncontrollable transitions of the plant should be bisimilar to those of the supervised plant, so that the reachable uncontrollable part of the former is indistinguishable from that of the latter. Note that the reachable uncontrollable part now contains the Markovian transitions as well, hence preserving the race condition that underpins the stochastic behavior. The controllable transitions of the supervised plant may only be simulated by the ones of the original plant, since some controllable transitions are suppressed by the supervisor. The stochastic behavior, represented implicitly by the Markovian transitions and the underlying race condition, is preserved due to lumping of the Markovian exit rates to equivalent states. Again, we emphasize that the supervisor does not contain any stochastic behavior as it should cater

only for proper disabling of controllable transitions.

Def. 4 Let P and S be given as above. We say that S is a supervisor for the plant P if $P \mid S <_U P$.

Def. 4 ensures that no uncontrollable actions have been disabled in the supervised plant, by including them in the bisimulation action set. Moreover, it ensures that the supervisor does not introduce any additional events, i.e., $A_S \subseteq A_P$. The stochastic behavior is correctly preserved up to Markovian lumping. In case P and S contain not stochastic behavior and they are deterministic, then Def. 4 coincides with the original definition of Ramadge and Wonham (1987).

Control Requirements

As given by the framework in Fig. 1, we separate the synthesis of a supervisor that ensures safe functioning of the supervised system from the analysis of the performance requirements of the system. Thus, the control requirements only concern functional behavior, and they abstract from the stochastic behavior in the system. Thus, we specify the control requirements by an IMC $R = (S_R, s_R, A_R, l_R, \emptyset, t_R)$. Since the supervised plant comprises stochastic behavior, we have to provide for an appropriate abstraction before relating it to the control requirements.

Def. 5 Let $I = (S, s_0, A, l, m, t)$ be an IMC. The time-abstracted IMC corresponding to I is given by $I^* = (S, s_0, A, l \cup l^*, \emptyset, t \cup t^*)$. For every state $s \in S$ and $s^* \in S$ such that $(s, p, s') \in m$, we put $(s, a, s^*) \in l^*$ if there exist $(s, p, s') \in m$, $(s', p', s'') \in m$, …, $(s''', p'', s'''') \in m$, and $(s'''', a, s^*) \in l$, and we put we put $s \in t^*$ if there exist $(s, p, s') \in m$, $(s', p', s'') \in m$, …, $(s''', p'', s^*) \in m$, and $s^* \in t$.

It should be clear that $S^* = S$ and $R^* = R$, since they do not contain any stochastic behavior. Now, we can define when the supervised plant satisfies the control requirement by using the time abstraction of Def. 5 and requiring that $(P \mid S)^* <_{\emptyset} R$. This relation states that the control requirements simulate the time-abstracted behavior of the supervised plant. The following theorem supports the separation of concerns, stating that every supervisor for the time abstracted plant is a supervisor for the original plant as well.

Thm. 2 If S is a supervisor for P, then S is a supervisor for P*, and vice versa, i.e., $P \mid S <_U P$ if and only if $(P \mid S)^* <_U P^*$.

The result of Thm. 2 relies on the fact that the supervisor does not ruin the stochastic behavior of the restriction of P. Most importantly, it enables us to employ standard synthesis tools, like Supremica, cf. Akesson et. al (2006), in the framework of Fig. 1, ensuring that the separate treatment of the control and performance requirements is consistent. In order to implement the abstractions and to derive the performance models, we built model transformation tools to the most prominent model checkers like PRISM, cf. Kwiatkowska et al. (2007), which are available for download from the author's website: http://sites.google.com/site/jasenmarkovski.

Next, we show the modeling and abstraction process on an illustrative case study dealing with supervisory coordination of resources.

Illustrative Case Study: Resource Allocation

We illustrate the modeling process on a simple resource allocation problem. We have processes P_i that request some resource with a given access rate, which is exponentially distributed, and once they are assigned the resource, they exploit it for some time, again exponentially distributed, and then they release it. We model such a process P_i as depicted in Fig. 2, where the initial state is denoted by an incoming arrow and Markovian transitions are dashed, where p_i and q_i are positive real numbers. The controllable events are req_i and asg_i, whereas the event rls_i is uncontrollable as we have no control over the execution of the processes.

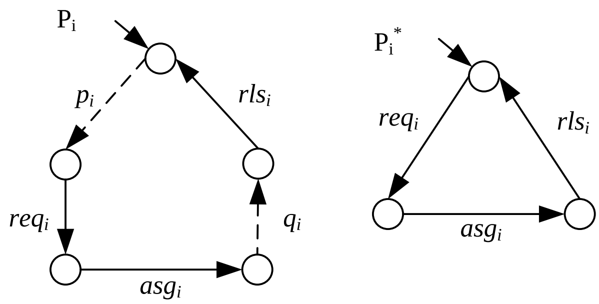

FIG. 2 MODEL OF A PROCESS REQUESTING A RESOURCE

Let us suppose that there are two processes that are requesting the same resource. Then, we need to coordinate the access to the resource. In this situation the plant P is given by the IMC $P = P_1 \mid P_2$. We can abstract from the stochastic behavior in the plant, which leads to the processes P_i^*, also depicted in Fig. 2. The qualitative control requirements that ensure proper access to the requested resource are modeled as given in Fig. 3.

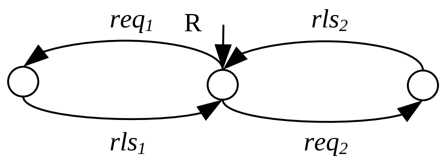

FIG. 3 MODEL OF THE CONTROL REQUIREMENTS

The control requirements state that once the resource has been assigned to one of the processes, then the other process has to wait its turn. We note that the assignment of the resource is not quantified by the control requirements, i.e., it is nondeterministic and it does not state anything about the actual behavior of the supervisory controller, but it only numbers all possibilities of supervision. We can synthesize a supervisor based on the abstracted plant $P^* = P_1^* \mid P_2^*$, which is depicted in Fig. 4.

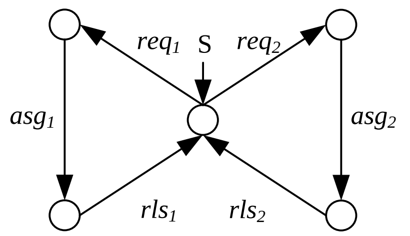

FIG. 4 MODEL OF THE SUPERVISORY CONTROLLER

It is not difficult to deduce that the synthesized supervisor for P* is also a supervisor for P. Now, we can form the stochastic supervised plant P ǀ S, which comprises nondeterministic behavior and abstract from the unnecessary labeled transitions, obtaining a Markov decision process, cf. Howard (1971), as depicted in Fig. 5.

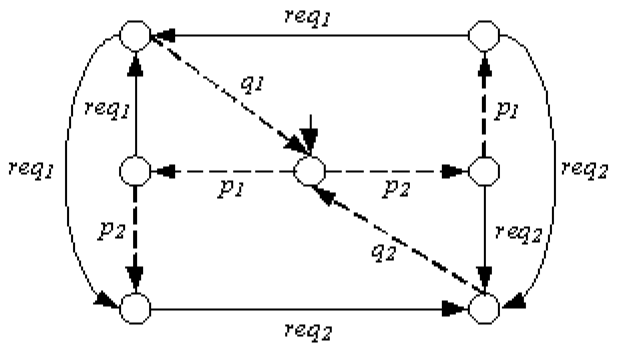

FIG. 5 PERFORMANCE MODEL OF THE SUPERVISED PLANT GIVEN IN THE FORM OF A MARKOV DECISION PROCESS

We note that the supervisor has a choice how to distribute the resource and assign it to the requesting processes. At this point, to optimize the supervision process, we can employ either the dynamic programming techniques from Bertsekas (2007) or the stochastic model checking approaches of Baier et al. (2004) in order to resolve the nondeterministic choice between the control actions req_1 and req_2. We can also make a design decision and, e.g., always prioritize P1, in which case we can derive and analyse a pure continuous-time Markov chain. In any case, the supervised plant is guaranteed to have the desired safe functionality of assigning the resource to one process at a time, which is the goal of the supervisory control synthesis.

Concluding Remarks

We proposed a process-theoretic approach to supervisory control theory of Interactive Markov Chains, an orthogonal extension of standard concurrency models with Markovian behavior, which we casted as a synthesis-centric model-based systems engineering framework. To this end, we propose to employ a behavioral relation termed Markovian partial bisimulation to capture the notion of controllability that correctly preserves the stochastic behavior. Our approach enabled us to abstract from the stochastic behavior in the plant and synthesize a supervisor using standard tools. Following the synthesis, we couple the supervisor with the stochastic model of the system, which is suitable for performance analysis after a suitable abstraction. We illustrated the modeling process by discussing resource allocation using supervisory coordination.

ACKNOLEDGEMENTS

This work is supported by by Dutch NWO project ProThOS, no. 600.065.120.11N124.

REFERENCES

Akesson, K., Fabian, M., Flordal, H., and Malik, R. Supremica - An integrated environment for verification, synthesis and simulation of discrete event systems. In Proceedings of WODES 2006, pages 384—385. IEEE, 2006.

Baeten, J. C. M., Basten, T., and Reniers, M. A. Process Algebra: Equational Theories of Communicating Processes, volume 50 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2010.

Baeten, J. C. M., Beek, D. A. van, Luttik, B., Markovski, and Rooda, J. E. A process-theoretic approach to supervisory control. In Proceedings of ACC 2011, pages 4496–4501. IEEE, 2011.

Baier, C., Grer, M., Leucker, M., Bollig, B., and Ciesinski, F. Controller synthesis for probabilistic systems. In Proceedings of IFIP TCS 2004, pages 493–506. Kluwer, 2004.

Baier, C., Haverkort, B. R., Hermanns, H., and Katoen, J.-P. Performance evaluation and model checking join forces. Communications of the ACM, 53(9):76–85, 2010.

Bertsekas, D. P. Dynamic Programming and Optimal

Control, volume 1 & 2. Athena Scientific, 2007.

Cassandras, C. and Lafortune, S. Introduction to Discrete Event Systems. Kluwer Academic Publishers, 2004.

Clark, A., Gilmore, S., Hillston, J., and Tribastone, M. Stochastic process algebras. In Formal Methods for Performance Evaluation, volume 4486 of Lecture Notes in Computer Science, pages 132–179. Springer, 2007.

Derisavi, S., Hermanns, H., and Sanders, W. H. Optimal state-space lumping in Markov chains. Information Processing Letters, 87(6):309 –315, 2003.

Garg, V. K., Kumar, R., and Marcus, S. I. A probabilistic language formalism for stochastic discrete-event systems. IEEE Transactions on Automatic Control, 44(2):280 – 293, 1999.

Glabbeek, R. J. van. The linear time–branching time spectrum I. Handbook of Process Algebra, pages 3–99, 2001.

Hermanns, H. and Katoen, J.-P. The how and why of Interactive Markov chains. In Proceedings of FMCO 2009, Lecture Notes in Computer Science, pages 311–337. Springer, 2010.

Heymann, M. and Lin F. Discrete-event control of nondeterministic systems. IEEE Transactions on Automatic Control, 43(1):3–17, 1998.

Hermanns, H., Herzog, U., and Katoen, J.-P. Process algebra for performance evaluation. Theoretical Computer Science, 274:43–87, 2002.

Hermanns, H. Interactive Markov Chains and the Quest For Quantified Quantity, volume 2428 of Lecture Notes of Computer Science. Springer, 2002.

Howard, R. A. Dynamic Probabilistic Systems, volume 1 & 2. John F. Wiley & Sons, 1971.

Kumar, R. and Garg, V. K. Control of stochastic discrete event systems: Synthesis. volume 3, pages 3299 –3304. IEEE, 1998.

Kumar, R. and Shayman, M. A. Nonblocking supervisory control of nondeterministic systems via prioritized synchronization. IEEE Transactions on Automatic Control, 41(8):1160–1175, 1996.

Kwiatkowska, M., Norman, G., and Parker, D. Stochastic model checking. In Formal Methods for Performance Evaluation, volume 4486 of Lecture Notes in Computer Science, pages 220–270. Springer, 2007.

Kwong, R. H. and Zhu L. Performance analysis and control of stochastic discrete event systems. In Feedback Control, Nonlinear Systems, and Complexity, volume 202 of Lecture Notes in Control and Information Sciences, pages 114–130. Springer, 1995.

Lawford, M. and Wonham, W. M.. Supervisory control of probabilistic discrete event systems. volume 1, pages 327–331. IEEE, 1993.

Markovski, J., Beek, D. A. van, Theunissen, R. J. M., Jacobs, K. G. M., and Rooda, J. E. A state-based framework for supervisory control synthesis and verification. In Proceedings of CDC 2010, pages 3481–3486. IEEE, 2010.

Markovski, J., Jacobs, K. G. M., Beek, D. A. van, L. Somers, J. A. M., and Rooda, J. E. Coordination of resources using generalized state-based requirements. In Proceedings of WODES 2010, pages 300–305. IFAC, 2010.

Markovski, J., Sokolova, A., Trcka, N., and Vink, E.P. de. Compositionality for Markov reward chains with fast and silent transitions. Peformance Evaluation, 66(8):435–452, 2009.

Markovski, J. Towards Supervisory Control of Interactive Markov Chains: Controllability. In Proceedings of ACSD 2011, pages 108–117. IEEE, 2011b.

Marsan, M. Ajmone, Balbo, G., Conte, G., Donatelli, S., and Franceschinis, G. Modelling with Generalized Stochastic Petri Nets. Wiley, 1995.

Overkamp, A. Supervisory control using failure semantics and partial specifications. IEEE Transactions on Automatic Control, 42(4):498–510, 1997.

Pantelic, V., Postma, S. M., and Lawford, M. Probabilistic supervisory control of probabilistic discrete event systems. IEEE Transactions on Automatic Control, 54(8):2013 – 2018, 2009.

Ramadge, P. J. and Wonham, W. M. Supervisory control of a class of discrete event processes. SIAM Journal on Control and Optimization, 25(1):206–230, 1987.

Rutten, J. J. M. M. Coalgebra, concurrency, and control. In Proceedings of WODES 2000, pages 300–305. Kluwer, 2000.

Schiffelers, R. R. H., Theunissen, R. J. M., Beek, D. A. van, and Rooda, J. E.. Model-based engineering of supervisory controllers using CIF. Electronic Communications of the EASST, 21:1–10, 2009.

Sengupta, R. and Lafortune, S. A deterministic optimal control theory for discrete event systems. volume 2, pages 1182 – 1187. IEEE, 1993.

Theunissen, R. J. M., Schiffelers, R. R. H., Beek, D. A. van, and Rooda, J. R.. Supervisory control synthesis for a patient support system. In Proceedings of ECC 2009, pages 1–6. EUCA, 2009.

Zhou C., Kumar, R., and Jiang S. Control of nondeterministic discrete-event systems for bisimulation equivalence. IEEE Transactions on Automatic Control, 51(5):754–765, IEEE, 2006.